PROJECT REPORT

NUTRITION ASSISTANT APPLICATION

Team ID: PNT2022TMID02196 **Batch:** B5-5M1E

TEAM LEADER:

Name: NANDITA SAJEEV **Register Number:** 2116190701120

TEAM MEMBERS:

Name: ABUTHAHIR **Register Number:** 2116190701006

Name: NITHISH KUMAR N **Register Number:**2116190701129

Name: PARTHIBAN M **Register Number:** 2116190701133

CONTENTS

1. INTRODUCTION

1.1 Project Overview

The project titled “Nutrition Assistant Application”, aims to provide a platform for users to make better nutritional choices and lead healthier lives. This is achieved by creating a convenient and easy-to-use application where users can upload images of the food they eat on a day-to-day basis and know about the nutritional value of the food. This can be especially useful for people with health concerns and who need to incorporate more of a certain nutrient in their food. It’s an easy way to educate people about the different nutrients in their foods as well.

1.2 Purpose

With the perpetual health craze, people in the technology era need an apt tool for them to understand nutrition in a convenient way. That’s where our project comes in handy. In this fast-paced world, people don’t have the time to focus on health and on-the-go meals or fast food is rising in popularity. This results in poor health among youth which can impact our future generation. The need of the hour is to make nutrition accessible and help people understand which nutrients are present in which foods.

2. LITERATURE SURVEY

2.1 Existing Problem

It's hard for people to eat a balanced diet, consisting of all the nutrients necessary for good health. Sometimes they end up overconsuming one type of food which makes them miss out on other nutrients.

2.2 References

1. **Development of a cloud-based solution for effective nutrition intervention in the management of lifestyle diseases.**

   Authors: Manju P George, C. A. Kalpana
   Year: 2020

   This paper proposes a system that aims to bridge the gap between clinical nutrition and the common man. For the purpose of prescribing therapeutic nutrition in clinical settings, a web-based application is being developed. The cloud-based solution would be able to figure out the nutritional needs and automatically direct first-line nutritional treatment to patients and clients. Additionally, it functions as an electronic medical and dietetic record, allowing for the planning of a customised nutrition counselling approach around the client's hectic schedule. One method is much simpler, and the client can speak with his or her personal nutritionist in a setting that suits them.

2. **Cloud-Based Meta learning System for Predictive Modeling of Biomedical Data**

   Authors: Milan VukiTeviT, Sandro RadovanoviT, Miloš MilovanoviT, and Miroslav MinoviT
   Year: 2013

   This research presented a cloud-based infrastructure for biomedical big data storage, processing, and predictive modelling. The meta-learning system is added to the existing service-based cloud architecture as a knowledge service that is data and model driven. We supported community-based data and

algorithm collecting as part of the suggested architecture because it is a crucial prerequisite for the high quality of meta-learning. Through a platform for the development and execution of distributed data mining processes and algorithms, this research field can advance and gain new value. Finally, we provide data- and model-driven decision help for choosing the optimal biomedical data processing techniques.

3. **DeepFood: Automatic Multi-Class Classification of Food Ingredients Using Deep Learning**

   Authors: Lili Pan, Samira Pouyanfar, Hao Chen, Jiaohua Qin
   Year: 2017

   This study suggests the DeepFood framework, which combines various deep feature sets, a number of feature selections, and an improved classifier known as SMO to automatically multi-class categorise food items using deep learning. The architecture is made to categorise small to medium-sized datasets, which is a highly common and essential task in practical applications.

4. **Study for Food Recognition System Using Deep Learning**

   Authors: Nareen O. M.Salim, Subhi R. M. Zeebaree, Mohammed A.M.Sadeeq A.H Radie
   Year: 2013

   This paper reviewed a significant number of recent articles on the APP on the deep learning of foodstuffs, and it went into detail about each article's structure, training methodology, and final assessment results of the deep learning for processing the food picture, spectrum, text, and other details. In terms of effectiveness, we compared deep learning to other widely used methodologies and found that, in these evaluated studies, deep learning produces superior results to other approaches. This essay discussed crucial Food Recognition. According to the literature study, food recognition is aided by a number of active mechanisms. The researchers have successfully used a variety of strategies and algorithms to accomplish this goal.

5. **Deep feature extraction technique based on Conv1D and LSTM network for food image recognition**

   Authors: **Sirawan Phiphitphatphaisit, Olarik Surinta**
   Year: **2021**

   The ResNet50+Conv1D-LSTM network was suggested in this study for precise food image identification. The reliable spatial features were first extracted. Second, the Conv1D network linked with the long short-term memory (LSTM) network, known as Conv1D-LSTM, employed robust characteristics as input data. The Conv1D-LSTM network's main job was to extract a temporal characteristic. The output of the Conv1D-LSTM was then converted into a probability distribution using the softmax algorithm.

6. **Automatic Fruits Detection Using Artificial Intelligence**

   Authors: **Tejswini Balpande, Nikita Dhothkar, Heena Satpute, Namrata Durbude**
   Year: **2020**

   In this study, we employ AI, which uses a camera to automatically determine the fruit's quality. The image can be scanned using a camera, after which image processing is carried out to identify the fruits' calories and other characteristics such as form, size, colour, and texture. In this study, image enhancement is the primary goal of image processing in order to reduce undesired noise and provide a better image. The camera's image is enhanced using image processing, which also identifies fruit characteristics like size, colour, and calories.

7. **Enhancing Cloud and Big Data Systems for healthy Food and Nutrition Information Systems Practice: A Conceptual Study**

   Authors: **P.K. Paul1, P.S. Aithal2, A. Bhuimali3**
   Year: **2019**

   This essay clarified the fundamentals of cloud computing, such as its fundamental attributes and functionalities. It also aided in our understanding of the primary difficulties associated with cloud computing and related technologies in the context of poor nations.

8. **Mobile cloud-based system recognizing nutrition and freshness of food image**

   Authors: **Diptee Kumbhar, Sarita Patil**
   Year: **2017**

   The framework presented in this research offers clients practical and clever methods that let them keep tabs on their calorie consumption and measure their food intake. Our system's food recognition method makes use of a cloud computing environment with classifier machine learning and a Naive Bayes training mechanism. Using image processing techniques, this device also verifies the fruit's freshness. The precision of the procedure used to measure calorie consumption is improved by this technology.

9. **Dietary Nutrition Cloud Platform Technology Based on Big Data**

   Author: **Muhammad Jmail**
   Year: **2021**

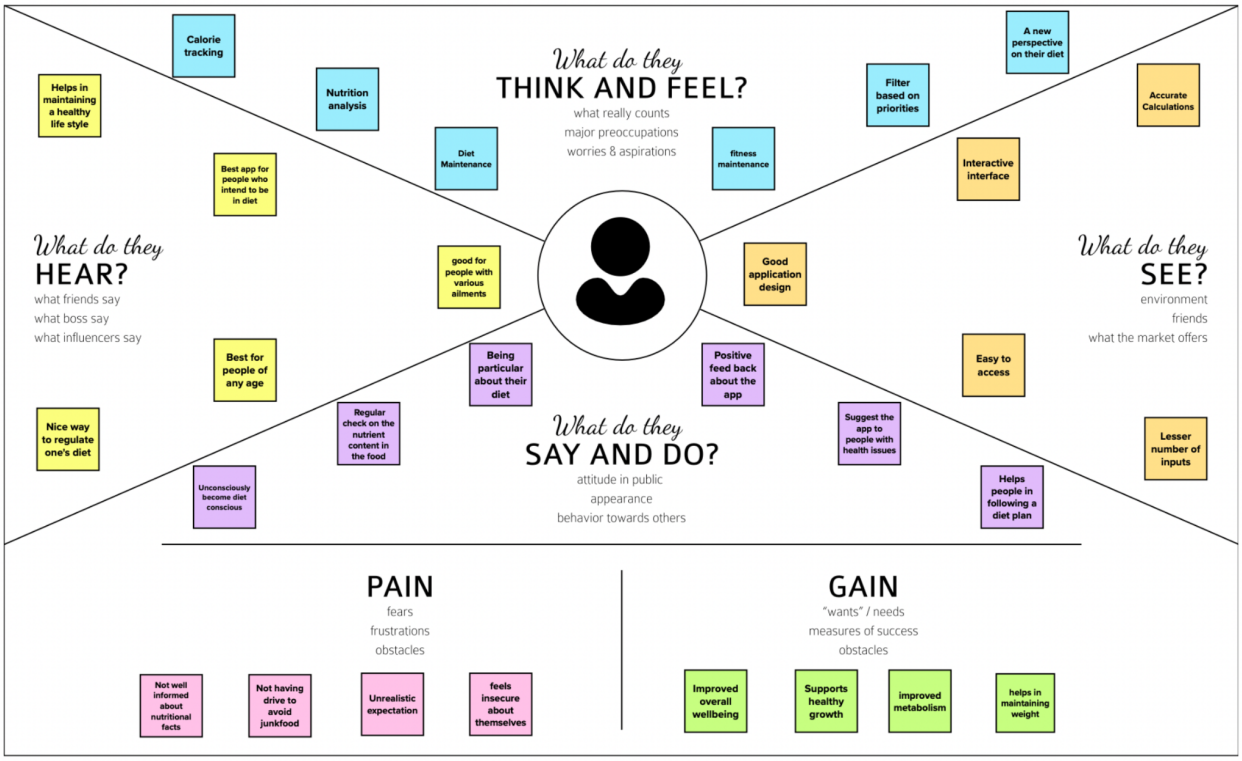
   In order to identify the relationship between dietary intake and disease, this paper analyses the characteristics of the food nutrition cloud platform, disease, and nutrition intake in great detail. It also identifies the drawbacks of the traditional association rule algorithm in the dietary nutrition cloud platform and suggests an improved immune algorithm based on clustering. The method can speed up association rule searches and can instantly locate the desired number of frequent item sets.

2.3 Problem Statement Definition

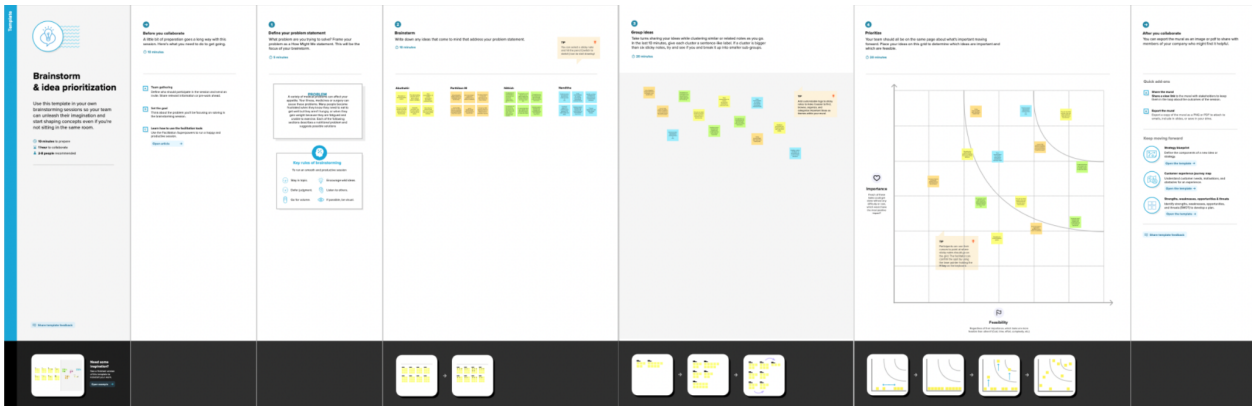
Obesity rates are rising alarmingly quickly as a result of people's lack of knowledge about appropriate eating practices, which reflects the hazards to their health. The simplest way to prevent obesity is for people to limit their daily calorie consumption by eating healthier meals. It's still not very convenient for people to use app-based nutrient dashboard systems, even though food packaging includes nutrition (and calorie) labels. These systems can analyse real-time images of a meal and analyse it for nutritional content, which can be very handy and improve dietary habits and subsequently help with maintaining a healthy lifestyle.

3. IDEATION AND PROPOSED SOLUTION

3.1 Empathy Map Canvas



3.2 Ideation and Brainstorming

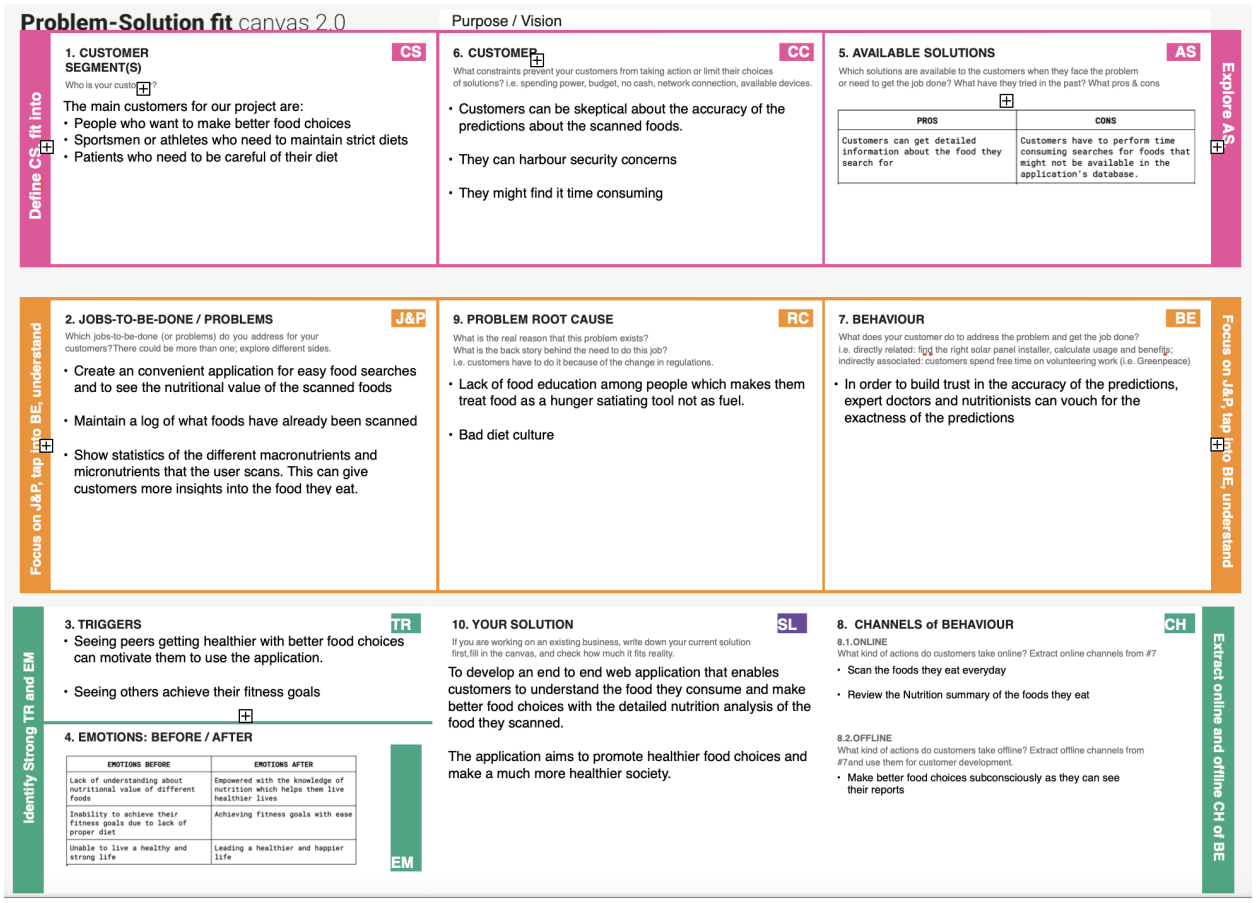


3.3 Proposed Solution

S.No.	Parameter	Description
●	Problem Statement (Problem to be solved)	● To develop web applications which automatically identify & quantify thousands of food categories and pair the food items with the relevant nutritional information for individuals to monitor and maintain the level of calorie intake.
●	Idea / Solution description	● This project aims at building a web application that automatically estimates food attributes such as ingredients and nutritional value by classifying the input image of food.Our method employs Clarify's AI-driven food detection model for accurate food identification and Food API's to give the nutritional value of the identified food.
●	Novelty / Uniqueness	● Plant Based Foods. ● Healthy, Safe, Living conditions. ● Using a global food matters Database. ● Patients to More Easily Monitor their caloric intake and dietary Pattern to aid in weight and disease Management.
●	Social Impact / Customer Satisfaction	● Subsidies And Piece Promotions on Healthy Food. ● Nutrition Focussed Food Banking. ● Corporate Social Responsibility initiatives to Increase dignified Access to Healthy food. ● Targeted Food Assistance Programs. ● It will help people with providing proper nutrition and helps in maintaining a healthy lifestyle.

•	Business Model (Revenue Model)	• Collaboration Other consultancies. • Customized Nutrition and consumption Requirements. • Health-Trade-Policy.
•	Scalability of the Solution	• Easily Access the Application. • Good Relationship.

3.4 Problem Solution Fit



4. REQUIREMENT ANALYSIS

4.1 Functional Requirements

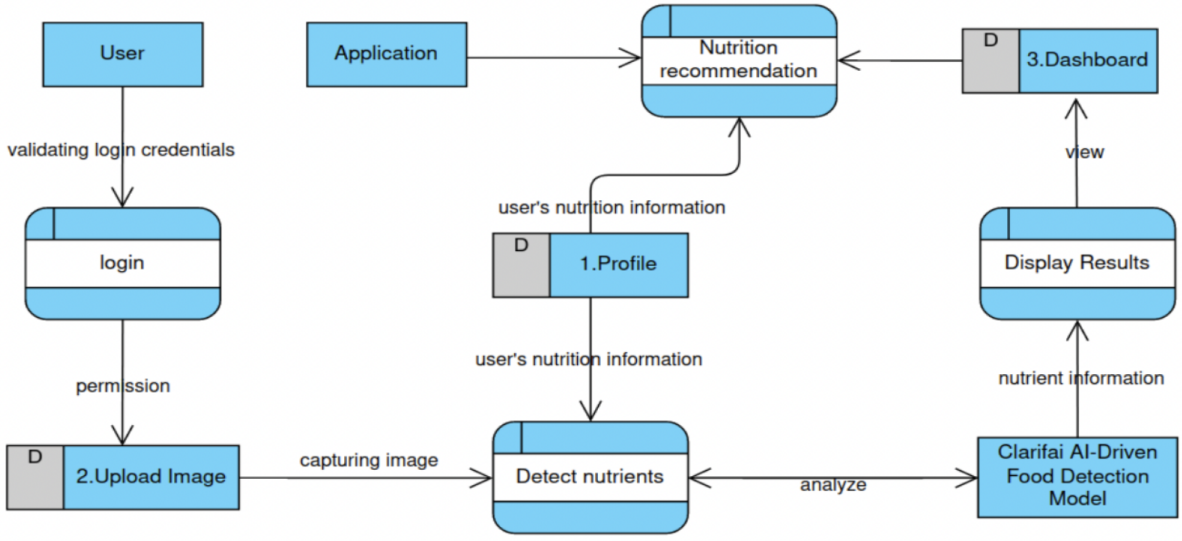
FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Form in the WebApp
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	Scanning the food	Access camera app or the gallery/files to get the image as input that we will pass to the Nutrition API to get predicted
FR-4	View the nutrition analysis of scanned food	The image after prediction, must display the amount of each nutrient it has.
FR-5	View food history	The user must be able to have a log of all the foods they have scanned on the app for at least the past 30 days
FR-6	Food Searching Option	The user must be able to search for common foods and find out the breakdown of its nutrients
FR-7	Diet analysis report	After the user has used the application consistently for a month, the application must analyse the nutritional value of the foods eaten and let the user know about the nutrients they are taking. It must also give them suggestions to achieve a more balanced diet like, foods that can be included in the diet, etc.

4.2 Non-Functional Requirements

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	Must have an interactive and user friendly user interface
NFR-2	**Security**	It must be secured with the proper username and password. Email and OTP verifications should also be done periodically.
NFR-3	**Reliability**	The system must always perform its operations promptly and reliably
NFR-4	**Performance**	User must have a compatible system and the application must deal with system's response time at different load conditions
NFR-5	**Availability**	The system must be available at all times with all it's functions working properly
NFR-6	**Scalability**	The system must be well equipped to handle huge workloads and still maintain performance requirements.

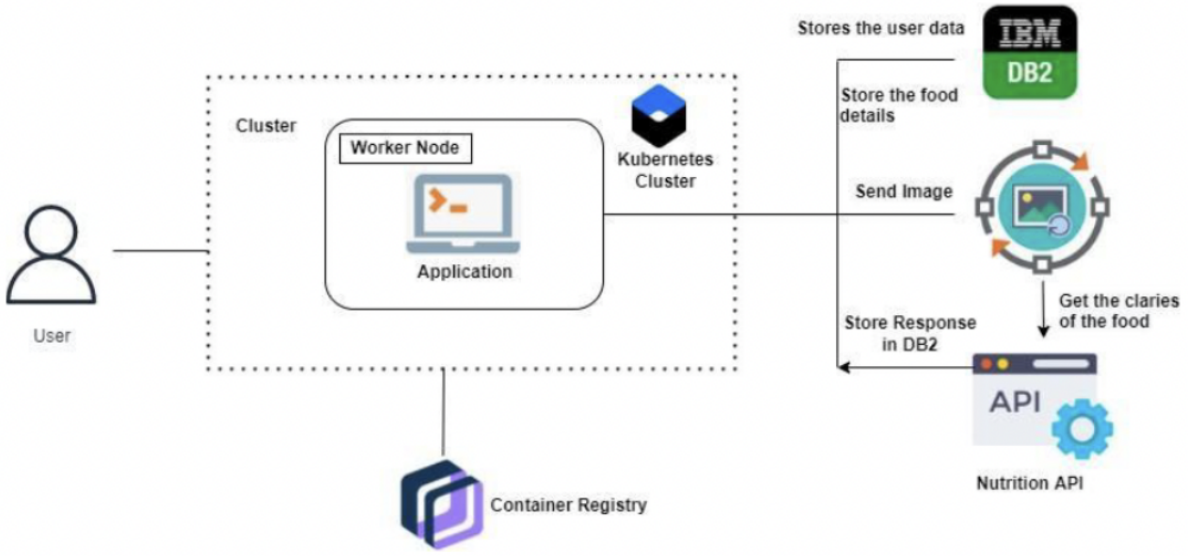
5. PROJECT DESIGN

5.1 Data Flow Diagram

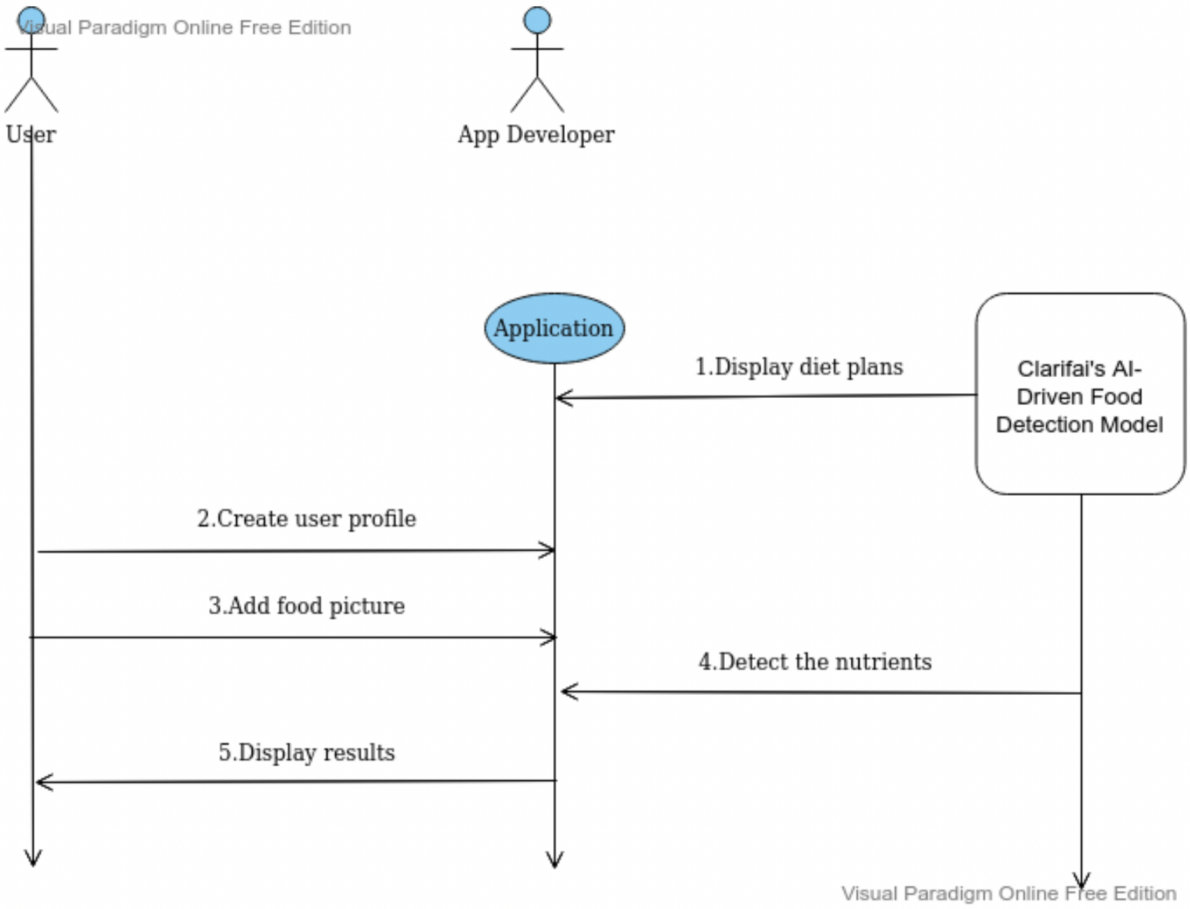


5.2 Solution and Technical Architecture

Technical Architecture



Solution Architecture



5.3 User Stories

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account / dashboard	High	Sprint-1
		USN-2	As a user, I will receive confirmation email once I have registered for the application	I can receive confirmation email & click confirm	High	Sprint-1
		USN-3	As a user, I can register for the application through Facebook	I can register & access the dashboard with Facebook Login	Low	Sprint-2
		USN-4	As a user, I can register for the application through Gmail	I can get notifications through Gmail	Medium	Sprint-1
	Login	USN-5	As a user, I can log into the application by entering email & password	I can access through my email and password	High	Sprint-1
	Dashboard	USN-6	As a user, I can upload or capture the food image using camera	I can use my camera or my files to upload image	High	Sprint-1
Customer (Web user)	Profile	USN-7	As a user, I can give my diet details	I can enter my diet information	High	Sprint-1
	Dashboard	USN-8	As a user, I can allow app to use my diet details to recommend nutrition diets	I can allow app to access my diet details	Medium	Sprint-2
Customer Care Executive	Application	USN-9	As a customer care executive, I can access customer's information and to solve their queries and issues	I can access I can access customer's information and to solve their queries and issues	Medium	Sprint-2
Administrator	Application	USN-10	As an administrator, I can manage and validate the customer's information	I can manage and validate the customer's information	High	Sprint-1
		USN-11	As an administrator, I can release updated versions of this application	I can release updated versions of this application	Medium	Sprint-2

6. PROJECT PLANNING AND EXECUTION

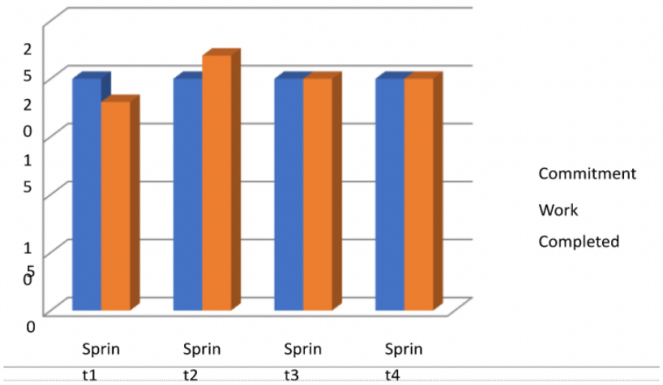
6.1 Sprint Planning and Scheduling

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	2	High	2
Sprint-1		USN-2	As a user, I will receive confirmation email once I have registered for the application	1	High	2
Sprint-2		USN-3	As a user, I can register for the application through Facebook	2	Low	2
Sprint-1		USN-4	As a user, I can register for the application through Gmail	2	Medium	2
Sprint-1	Login	USN-5	As a user, I can log into the application by entering email & password	1	High	2
Sprint-1	Dashboard	USN-6	As a user, I can upload or capture the food image using camera	1	High	2
Sprint-2		USN-7	As a user, I can allow app to use my diet details to recommend nutrition diets	1	Low	2
Sprint-1	Administration	USN-8	As an administrator, I can manage and validate the customer's information	2	High	2
Sprint-1		USN-9	As an administrator, I can release updated versions of this application	2	Medium	4
Sprint-2	Customer Care	USN-10	As a customer care executive, I can solve the queries of users	1	Low	2

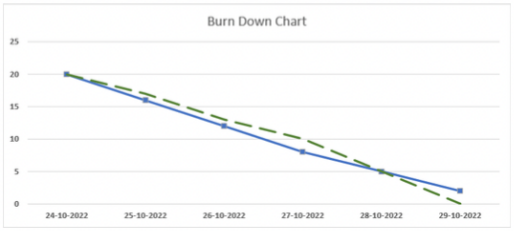
6.2 Sprint Delivery Schedule

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022	20	30 Oct 2022
Sprint-2	20	6 Days	01 Nov 2022	06 Nov 2022	20	07 Nov 2022
Sprint-3	20	6 Days	09 Nov 2022	14 Nov 2022	20	15 Nov 2022
Sprint-4	20	6 Days	17 Nov 2022	22 Nov 2022	20	23 Nov 2022

Velocity chart:

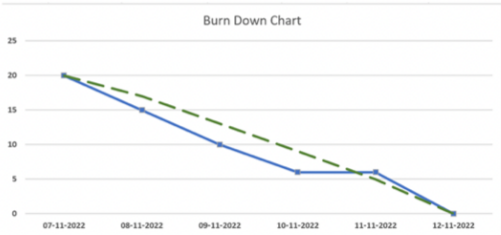


Burn down chart:



Sprint 1

Sprint 2



Sprint 3

Sprint 4

7. CODING & SOLUTIONING

7.1 Feature 1

Python Flask

Python Flask is used to develop chatbot applications using python. Flask is mainly used to render and integrate the nutrition assistant application in the browser by providing API. By running the python application, the suitable server domain link is obtained and run in the browser.

HTML

The HTML and CSS is used to design the overall nutrition assistant application's UI**.** HTML is used to add UI components and CSS is used to add style to those components.

Build PYTHON FLASK Code:

APP.PY

```
from flask import Flask, render_template, redirect, url_for, request, flash, escape, session

from flask_wtf import FlaskForm

from wtforms import StringField, PasswordField, EmailField

from wtforms.validators import InputRequired, Length, Email, EqualTo

import ibm_db

import time

import openapi_client

from com.spoonacular import misc_api

from openapi_client.model.image_analysis_by_url200_response import
ImageAnalysisByURL200Response

from pprint import pprint

from flask import Flask,redirect,url_for,render_template,request

import ibm_boto3

from ibm_botocore.client import Config, ClientError

import json
```

```
import os

from dotenv import load_dotenv

load_dotenv()

COS_ENDPOINT="https://s3.jp-tok.cloud-object-storage.appdomain.cloud"

COS_API_KEY_ID= os.getenv("COSAPI")

COS_INSTANCE_CRN= os.getenv("COSINSTANCE")

# Create resource https://s3.ap.cloud-object-storage.appdomain.cloud

cos = ibm_boto3.resource("s3",

    ibm_api_key_id=COS_API_KEY_ID,

    ibm_service_instance_id=COS_INSTANCE_CRN,

    config=Config(signature_version="oauth"),

    endpoint_url=COS_ENDPOINT

)

DB_HOSTNAME = os.getenv("DB_HOSTNAME")

DB_PORT = os.getenv("DB_PORT")

DB_USERNAME = os.getenv("DB_USERNAME")

DB_PASS = os.getenv("DB_PASS")

#conn =
ibm_db.connect(f"DATABASE=bludb;HOSTNAME={DB_HOSTNAME};PORT={DB_PORT};SECURITY=SSL;SSLServerC
ertificate=DigiCertGlobalRootCA.crt;UID={DB_USERNAME};PWD={DB_PASS}",'','')

conn =
ibm_db.connect("DATABASE=bludb;HOSTNAME=3883e7e4-18f5-4afe-be8c-fa31c41761d2.bs2io90l08kqb1od
8lcg.databases.appdomain.cloud;PORT=
31498;SECURITY=SSL;SSLServerCertificate=DigiCertGlobalRootCA.crt;UID=sxq13787;PWD=h4bJWERMtK7
2WIkp",'', '')

print(conn)

app = Flask(__name__)

app.config['SECRET_KEY'] = os.getenv("SECERT_KEY")

class LoginForm(FlaskForm):

    email = EmailField("email", validators=[InputRequired("Email is required"), Email()])

    password = PasswordField("password", validators=[InputRequired("Password is required")])
```

```
class RegisterForm(FlaskForm):

    username = StringField("username", validators=[InputRequired("Username is required")])

    email = EmailField("email", validators=[InputRequired("Email is required"), Email()])

    pass1 = PasswordField("pass1", validators=[InputRequired("Password is required"),
EqualTo('pass2', message="Passwords must match"), Length(min=4, max=30, message="Length must
be between 4 and 30")])

    pass2 = PasswordField("pass2")

class ForgetPassword(FlaskForm):

    email = EmailField("email", validators=[InputRequired("Email is required"), Email()])

@app.route("/")

def home():

    username = request.cookies.get('username')

    return render_template("home.html", username=username)

@app.route('/login', methods=['GET', 'POST'])

def login():

    form = LoginForm()

     if request.method=='POST' and form.validate_on_submit():

        email = request.form['email']

        password = request.form['password']

        sql = f"SELECT * FROM user WHERE EMAIL='{escape(email)}'"

        stmt = ibm_db.exec_immediate(conn, sql)

        dic = ibm_db.fetch_both(stmt)

        if not dic or password != dic['PASSWORD']:

            flash("Incorrect email or password", "error")

            return redirect(url_for('login'))

        session['username'] =  dic['USERNAME']

        return redirect(url_for('home'))

    Else:
```

```
return render_template("login.html", form=form)

@app.route('/signup', methods=['GET', 'POST'])

def signup():

    form = RegisterForm()

    if request.method=='POST' and form.validate_on_submit():

        username = request.form['username']

        email = request.form['email']

        pass1 = request.form['pass1']

        sql = f"SELECT * FROM user WHERE EMAIL='{escape(email)}'"

        stmt = ibm_db.exec_immediate(conn, sql)

        dic = ibm_db.fetch_both(stmt)

        if dic:

            flash("User with the email already exist", "error")

            return redirect(url_for('login'))

        sql = "INSERT INTO user(username,email,password) VALUES (?, ?, ?)"

        prep_stmt = ibm_db.prepare(conn, sql)

        ibm_db.bind_param(prep_stmt, 1, username)

        ibm_db.bind_param(prep_stmt, 2, email)

        ibm_db.bind_param(prep_stmt, 3, pass1)

        ibm_db.execute(prep_stmt)

        flash("Registration Successful", "success")

        response = redirect('/login',307)

        return response

    else:

        return render_template("register.html", form=form)

@app.route('/forgot', methods=['GET', 'POST'])

def forgot_password():
```

```
    form = ForgetPassword()
    if request.method=='POST' and form.validate_on_submit():
        email = request.form['email']
        sql = f"SELECT * FROM user WHERE EMAIL='{escape(email)}'"
        stmt = ibm_db.exec_immediate(conn, sql)
        dic = ibm_db.fetch_both(stmt)
        if dic:
            flash("Email has been sent if user exist", "success")
            return redirect(url_for('forgot_password'))
       return render_template("forgot_password.html", form=form)
    return render_template("forgot_password.html", form=form)
  @app.route('/logout', methods=['GET', 'POST'])
def logout():
    session.pop('username')
    return redirect(url_for('home'))
@app.route('/pictures')
def index():
    files = get_bucket_contents('flask-app-2k22')
    return render_template('index.html', files = files)
@app.route('/uploader', methods = ['GET', 'POST'])
def upload():
   if request.method == 'POST':
       bucket=request.form['bucket']
       name_file=request.form['filename']
       f = request.files['file']
       multi_part_upload(bucket,name_file,f.filename)
       sql = f"INSERT INTO imagedetails(img_link) VALUES(?)"
```

```
imagelink="https://flask-app-2k22.s3.jp-tok.cloud-object-storage.appdomain.cloud/"+name_file
        print(imagelink)
        prep_stmt = ibm_db.prepare(conn, sql)
        ibm_db.bind_param(prep_stmt, 1, imagelink)
        ibm_db.execute(prep_stmt)
        sql = f"SELECT ID FROM imagedetails WHERE img_link='{escape(imagelink)}'"
        stmt = ibm_db.exec_immediate(conn, sql)
        image_id = ibm_db.fetch_both(stmt)
        nutitionapi(imagelink,image_id)
        return redirect("/foodinfo", code=307)
if request.method == 'GET':
        return render_template('upload.html')
def get_item(bucket_name, item_name):
    print("Retrieving item from bucket: {0}, key: {1}".format(bucket_name, item_name))
    try:
        file = cos.Object(bucket_name, item_name).get()
print("File Contents: {0}".format(file["Body"].read()))
    except ClientError as be:
        print("CLIENT ERROR: {0}\n".format(be))
    except Exception as e:
        print("Unable to retrieve file contents: {0}".format(e))
def get_bucket_contents(bucket_name):
    print("Retrieving bucket contents from: {0}".format(bucket_name))
    try:
        files = cos.Bucket(bucket_name).objects.all()
        files_names = []
```

```
        for file in files:
            files_names.append(file.key)
            print("Item: {0} ({1} bytes).".format(file.key, file.size))
        return files_names
    except ClientError as be:
        print("CLIENT ERROR: {0}\n".format(be))
    except Exception as e:
        print("Unable to retrieve bucket contents: {0}".format(e))
def multi_part_upload(bucket_name, item_name, file_path):
    try:
        print("Starting file transfer for {0} to bucket: {1}\n".format(item_name,
bucket_name))
        # set 5 MB chunks
        part_size = 1024 * 1024 * 5
       # set threadhold to 15 MB
        file_threshold = 1024 * 1024 * 15
       # set the transfer threshold and chunk size
        transfer_config = ibm_boto3.s3.transfer.TransferConfig(
            multipart_threshold=file_threshold,
            multipart_chunksize=part_size
        )
        # the upload_fileobj method will automatically execute a multi-part upload
        # in 5 MB chunks for all files over 15 MB
        with open(file_path, "rb") as file_data:
            cos.Object(bucket_name, item_name).upload_fileobj(
                Fileobj=file_data,
                Config=transfer_config
```

```
 print("Transfer for {0} Complete!\n".format(item_name))

 except ClientError as be:

        print("CLIENT ERROR: {0}\n".format(be))

    except Exception as e:

        print("Unable to complete multi-part upload: {0}".format(e))

def nutitionapi(imagelink,image_id):

    configuration = openapi_client.Configuration(

        host = "https://api.spoonacular.com"

    )

    configuration.api_key['apiKeyScheme'] = os.getenv("NUTRITIONAPI")

    with openapi_client.ApiClient(configuration) as api_client:

        api_instance = misc_api.MiscApi(api_client)

        image_url =imagelink

    try:

        api_response = api_instance.image_analysis_by_url(image_url)

        pprint(api_response)

        y =api_response

        cal= y["nutrition"]["calories"]["value"]

        print(cal)

        Carb= y["nutrition"]["carbs"]["value"]

        fat= y["nutrition"]["fat"]["value"]

        protein= y["nutrition"]["protein"]["value"]

        name=y["category"]["name"]

        image=image_id["ID"]

        sql = f"INSERT INTO nutritiondetails1(calories,carbs,fat,protein,ref_id,name)
VALUES('{escape(cal)}','{escape(Carb)}','{escape(fat)}','{escape(protein)}','{escape(image)}'
,'{escape(name)}')"
```

```
        prep_stmt = ibm_db.prepare(conn, sql)

        ibm_db.execute(prep_stmt)

        flash("Successful db operation", "success"

    except openapi_client.ApiException as e:

        print("Exception when calling MiscApi->image_analysis_by_url: %s\n" % e)

@app.route('/foodinfo', methods = ['GET', 'POST'])

def test():

    sql = f"SELECT * FROM imagedetails "

    stmt = ibm_db.exec_immediate(conn, sql)

    pic = ibm_db.fetch_both(stmt)

    pics=[]

    while pic != False:

     x=[pic["IMG_LINK"],pic["ID"]]

     pics.append(x)

     print(pic)

     pic = ibm_db.fetch_both(stmt)

     print(pics)

    return render_template('index.html', files = pics)

@app.route('/nutritioninfo/<id>', methods = ['GET', 'POST'])

def test1(id):

    print(id)

    sql = f"SELECT * FROM nutritiondetails1,imagedetails where
nutritiondetails1.ref_id=imagedetails.id and ref_id='{escape(id)}'"

    stmt = ibm_db.exec_immediate(conn, sql)

    pic = ibm_db.fetch_both(stmt)

    print(pic)

    return render_template('foodinfo.html', files = pic)
```

```
if __name__ == '__main__':

    app.run(debug=True)
```

REGISTER.HTML

```
{% extends "base.html" %}


{% from "_render_field.html" import render_error_field %}


{% block head %}

<title>Registration</title>

<link rel="stylesheet" href="{{url_for('static', filename='css/login_style.css')}}">

{% endblock %}


{% block body %}

   <div class="container h-100">

    <div class="wrapper">

        <div class="card">

            <form class="d-flex flex-column" action="{{ url_for('signup') }}" method="POST"
novalidate>

                {{ form.csrf_token }}

                <div class="h3 text-center text-white">Sign Up</div>

                <div class="d-flex align-items-center input-field my-3 mb-4">

                    <span class="far fa-user p-2"></span>

                    {{ form.username(class="form-control", placeholder="Username") }}

                </div>

                {{ render_error_field(form.username) }}
```

```
                <div class="d-flex align-items-center input-field mb-4">
                    <span class="far fa-envelope p-2"></span>
                    {{ form.email(class="form-control", placeholder="Email") }}
                </div>
                {{ render_error_field(form.email) }}

                <div class="d-flex align-items-center input-field mb-4">
                    <span class="fas fa-lock p-2"></span>
                    {{ form.pass1(class="form-control", placeholder="New Password", id='pwd')
}}
                    <button type="button" class="btn" onclick="showPassword()">
                        <span class="fas fa-eye-slash"></span>
                    </button>
                </div>
                {{ render_error_field(form.pass1) }}

                <div class="d-flex align-items-center input-field mb-4">
                    <span class="fas fa-lock p-2"></span>
                    {{ form.pass2(class="form-control", placeholder="New Password",
id='pwd1') }}
                    <button type="button" class="btn" onclick="Password()">
                        <span class="fas fa-eye-slash"></span>
                    </button>
                </div>
                {{ render_error_field(form.pass2) }}

                <div class="my-3">
                    <input type="submit" value="Register" class="btn btn-primary">
```

```
                </div>


                <div class="mb-3">
                    <span class="text-light-white">Already have an account?</span>
                    <a href="/login">Login</a>
                </div>


            </form>


        </div>
    </div>
</div>
{% endblock %}
```

HOME.HTML

```
{% extends "base.html" %}
{% block head %}
       <title>Nutrify</title>
    <link rel="stylesheet" type="text/css" href="{{url_for('static',
filename='css/indexstyle.css')}}">
{% endblock %}
{% block body %}
<header>
    <nav class="navbar navbar-expand-lg navigation-wrap">
        <div class="container">
            <a class="navbar-brand" href="/">Nutrition Assistant Application</a>
            <button class="navbar-toggler" type="button" data-bs-toggle="collapse"
data-bs-target="#navbarNav"
```

```
                aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle
navigation">
                <span class="navbar-toggler-icon"></span>
            </button>
            <div class="collapse navbar-collapse" id="navbarText">

                <ul class="navbar-nav ms-auto mb-2 mb-lg-0">
                    <li class="nav-item">
                        <a class="nav-link" aria-current="page" href="/">Home</a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link" aria-current="page" href="/foodinfo">Food
Info</a>
                    </li>
                    {% if session['username'] %}
                    <li class="nav-item">
                        <a class="nav-link" aria-current="page" href="/uploader">Upload
Images</a>
                    </li>
                    {% endif %}
                    <li class="nav-item">
                        <a class="nav-link" aria-current="page" href="/#aboutus">About Us</a>
                    </li>
                    {% if session['username'] %}
                    <li>
                        <a class="btn btn-outline-danger ms-2" href="{{ url_for('logout')
}}">Logout</a>
                    </li>
                    {% else %}
```

```
                        <li>

                            <a class="btn btn-outline-danger ms-2" href="/login"
role="button">Login</a>

                        </li>

                        <li>

                            <a class="btn btn-primary ms-4" href="/signup"
role="button">Register</a>

                        </li>

                    {% endif %}

                </ul>

            </div>

        </div>

    </nav>

</header>

<section id="home">

  <div class="test">

    <div class="main">

    </div>

    <div class="textbox">

        {% if session['username'] %}

        <h1>Welcome <span style="color:#91eb48;">{{ session['username'] | title
}}</span></h1>

        <!-- <a href="{{ url_for('logout') }}">Logout</a> -->

    {% else %}

        <h1>Welcome to  <span style="color:#40a008;">Nutrify</span></h1>

    {% endif %}

    <h4>All groundwork of happiness starts from health.</h4>

    </div>
```

```
  </div
</section>
<section class="feature section-padding" id="feature">
       <div class="container-fluid px-0 top-banner1">
       <div class="container">
              <div class="feature-heading" style="visibility: visible; animation-name: zoom">
                     <h2><span style="color:#40a008;">Nutrify</span>'s Main Features</h2>
              </div>
              <div class="serv-field row mt-4">
                <div class="col-12 col-md-6 col-lg-4 text-center">
                    <div class="serv-box">
                        <div class="icon">
                            <i class="bi bi-stopwatch-fill"></i>
                        </div>
                        <h4>60 seconds</h4>
                        <p>Professional grade weekly meal plan creation in under 60
seconds.</p>
                    </div>
                </div>
                <div class="col-12 col-md-6 col-lg-4 text-center">
                    <div class="serv-box">
                        <div class="icon">
                            <i class="bi bi-menu-up"></i>
                        </div>
                        <h4>3,000+ variations</h4>
                        <p>Such as smoothies, soups, main dishes and so much more.</p>
                    </div>
```

```
                </div>
                <div class="col-12 col-md-6 col-lg-4 text-center">
                    <div class="serv-box">
                        <div class="icon">
                            <i class="bi bi-cloud-arrow-down-fill"></i>
                        </div>
                        <h4>1-Click Download</h4>
                        <p>Easily access from any device &amp;amp; share meal plans from any
device.</p>
                    </div>
                </div>
            </div>
        </div>
</div>
</section>
<script type="text/javascript" >
        let nav=document.querySelector(".navigation-wrap");
        window.onscroll = function(){
                if(document.documentElement.scrollTop > 20){
                        nav.classList.add("scroll-on");
                }else{
                        nav.classList.remove("scroll-on");
                }
        }
</script>
<section class="feature section-padding" id="trackfood">
        <div class="container-fluid px-0 top-banner1">
```

```
        <div class="container">

               <div class="feature-heading" style="visibility: visible; animation-name: zoom">

                      <h2>KNOW THE NUTRIENTS PRESENT IN YOUR FOOD</h2>

               </div>

               <div class="serv-field row mt-4">

                 <div class="col-12 col-md-6 col-lg-4 text-center">

                     <div class="serv-box">

                         <div class="icon">

                             <i class="bi bi-camera-fill"></i>

                         </div>

                        <h1 style="font-size: xxx-large;

        color: red;

        font-family: 'Gill Sans', 'Gill Sans MT', Calibri, 'Trebuchet MS',
sans-serif;">01</h1>

                         <p>Click a Picture of Your Food</p>

                     </div>

                 </div>

                 <div class="col-12 col-md-6 col-lg-4 text-center">

                     <div class="serv-box">

                         <div class="icon">

                             <i class="bi bi-cloud-arrow-up-fill"></i>

                         </div>

                        <h1 style="font-size: xxx-large;

        color: red;

        font-family: 'Gill Sans', 'Gill Sans MT', Calibri, 'Trebuchet MS',
sans-serif;">02</h1>

                         <p>Upload the Picture</p>

                     </div>
```

```
                </div>

                <div class="col-12 col-md-6 col-lg-4 text-center">

                    <div class="serv-box">

                        <div class="icon">

                            <i class="bi bi-card-list"></i>

                        </div>

                       <h1 style="font-size: xxx-large;

        color: red;

        font-family: 'Gill Sans', 'Gill Sans MT', Calibri, 'Trebuchet MS',
sans-serif;">03</h1>

                        <p>Know the Nutrients Present</p>

                    </div>

                </div>

            </div>

            <br>

            <br>

            <center>

                {% if session['username'] %}

    <a href="/uploader" class="btn btn-outline-primary" role="button" aria-pressed="true">
TRY NOW </a>

    {% endif %}

  </center>

       </div>

</div>

</section>

<footer id="aboutus">

        <div class="container">

            <div class="row pb-4">
```

```
                <div class="foot-info col-12 col-md-6 col-lg-9">

                    <a class="foot-logo" href="#home">

                        Nutri<span style="color:#40a008;" >fy</span>

                    </a>

                    <div class="mail">

                        <i class="fas fa-envelope"></i>

                        <a
href="mailto:nandita.sajeev.2019.cse@rajalakshmi.edu.in">nandita.sajeev.2019.cse@rajalakshmi.
edu.in</a>

                    </div>

                    <div class="mail">

                       <p>Contact Us: +91 12345 67890</p>

                    </div>

                </div>

            </div>

    </footer>

{% endblock %}
```

7.2 Feature 2

Upload Image

This page allows users to upload images of food in order to get the nutrition information of it.

UPLOADIMG.HTML

```
{% extends "base.html" %}

{% block head %}
        <title>Nutrify</title>
    <link rel="stylesheet" type="text/css" href="{{url_for('static',
filename='css/indexstyle.css')}}">
```

```
       <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/css/bootstrap.min.css"
rel="stylesheet"
integrity="sha384-Zenh87qX5JnK2Jl0vWa8Ck2rdkQ2Bzep5IDxbcnCeuOxjzrPF/et3URy9Bv1WTRi"
crossorigin="anonymous">
    <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/js/bootstrap.bundle.min.js"
integrity="sha384-OERcA2EqjJCMA+/3y+gxIOqMEjwtxJY7qPCqsdltbNJuaOe923+mo//f6V8Qbsw3"
crossorigin="anonymous"></script>
{% endblock %}

{% block body %}
<div class="row">

       <div class="col-lg-6">
       <div class="upload card">
              <h1>Upload Food Image</h1>
       <form class="uploaderform" action = "/uploader" method = "POST"
       enctype = "multipart/form-data">

         <div class="mb-3">
              <label for="exampleInputname" class="form-label">Name</label>
              <input placeholder="Enter bucket name" type="text" name="bucket"
class="form-control" id="exampleInputname" aria-describedby="emailHelp">
         </div>
         <div class="mb-3">
              <label for="exampleInputfilename" class="form-label">Filename</label>
              <input placeholder="Enter file name" name="filename" type="text"
class="form-control" id="exampleInputfilename">
         </div>
         <div class="mb-3">
                <input  class="form-control" type = "file" name = "file" />

         </div>
         <button type="submit" class="btn btn-primary">Submit</button>
       </form>
</div>
</div>

<div class="col-lg-6">
<img src="/static/images/logo.png" alt="">
</div>

</div>
{%  endblock %}
```

7.3 Feature 3

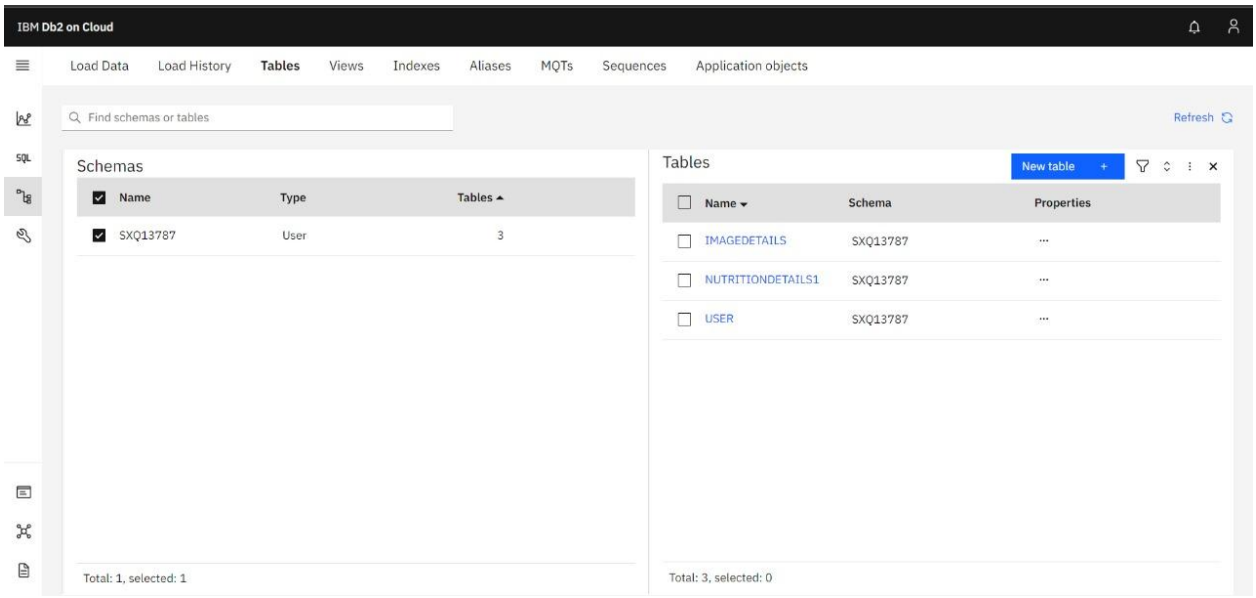
View History of Items

The users can view the nutritional information of all the past foods they have uploaded pictures of

FOODINFO.HTML

```
{% extends "base.html" %}

{% block head %}
       <title>Nutrify</title>
    <link rel="stylesheet" type="text/css" href="{{url_for('static',
filename='css/indexstyle.css')}}">
{% endblock %}

{% block body %}
<br>
<br>
<br>

<div class="foodinfomain">
<div class="foodinfo card">
    <div class="row">
        <div class="col-lg-6">
            <img class="fo" src="{{files[8]}}" alt="">

        </div>
<div class="col-lg-6">

    <p>Food item: {{files[1]}}</p>
    <hr>
    <p>Calories: {{files[2]}} Kcal</p>
    <hr>
    <p>Carbs: {{files[3]}} Gm</p>
    <hr>
    <p>Fat: {{files[4]}} Gm</p>
    <hr>
    <p>Protein: {{files[5]}} Gm</p>

</div>
    </div>

</div>
</div>

{% endblock %}
```

7.5 Database Schemas



8. TESTING

8.1 Test Cases

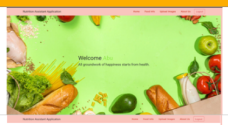
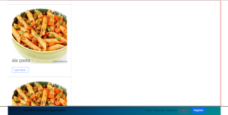
	Test Scenarios
1	Verify if the user is able to open and view the homepage
2	Verify if the user is able to interact with the elements in the homepage
3	Verify if the user is able to navigate to the other pages of the application from the homepage
	Upload Image Page Actions
1	User is able to upload image
2	User is able to submit the image and obtain results
	View History of Items Related Actions
1	User is able to view all past uploaded images
2	User is able to see the nutritional breakdown of the previously uploaded images
	User is able to log in and sign up
1	User is able to create an account and log in

8.2 User Acceptance Testing

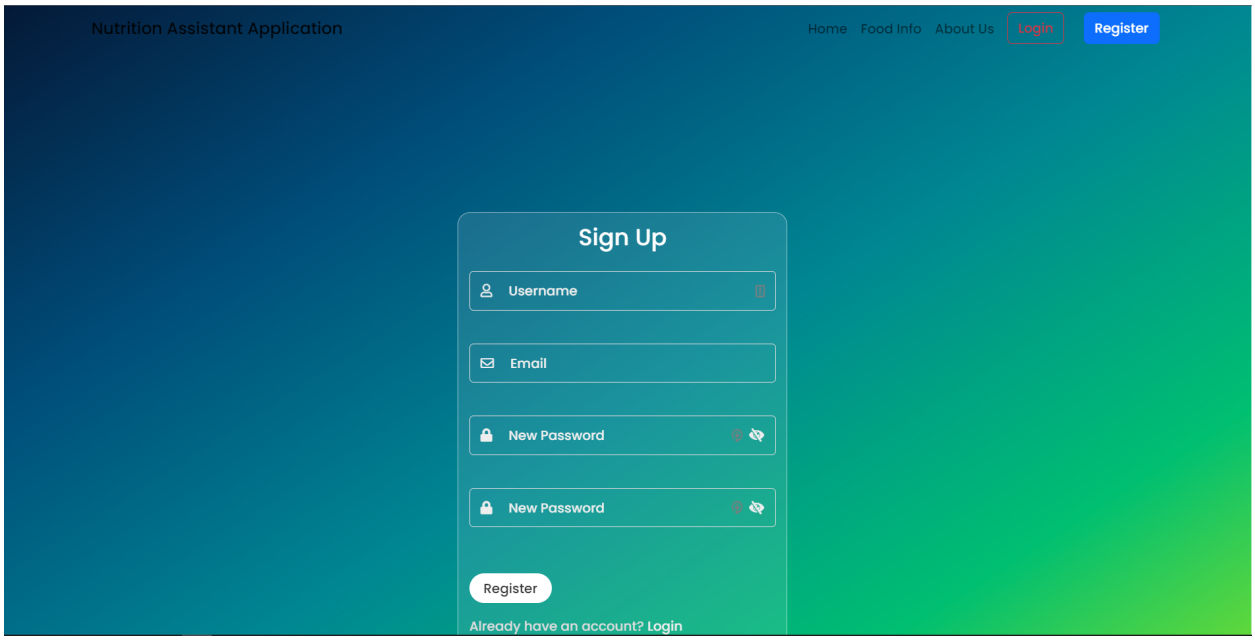
Test Case ID	Feature Type	Component	Test Scenario	Pre-requisite	Steps to Execute	Test Data	Expected Result	Actual Result	Status	Comments	TC for automation	Bug ID	Executed By
Homepage TC 01	UI	Homepage	Verify if the user is able to open and view the homepage	None	1. Click on URL and go to the homepage	URL Link	Homepage is viewable	Working as expected	Pass	-	N	-	Nandita S
Homepage TC 02	Functional	Homepage	Verify if the user is able to interact with the elements in the homepage	Homepage is accesible	1. Click on the various elements of the page and see if its working	Homepage	Elements Work	Working as expected	Pass	-	N	-	Nithish Kumar N
Homepage TC 03	Functional	Homepage	Verify if the user is able to navigate to other pages from the homepage	Homepage is accesible	1. Click on the various links of the page and see if its working	Homepage	We can navigate	Working as expected	Pass	-	N	-	Abuthahir
Upload Image TC 01	UI	Upload Image Page	User is able to upload images	Page is accessible	1. Click on upload image button and see if its working	Upload Image Page	We can upload	Working as expected	Pass	-	N	-	Parthiban
Upload Image TC 02	Functional	Upload Image Page	User is able to submit images and get results	Page is accessible	1. See if the uploaded images yield results	Upload Image Page	We can see results	Working as expected	Pass	-	N	-	Nandita S Abuthahir
View History TC 01	UI	View History Page	User is able to view past uploaded images	Page is accessible	1. See if the uploaded images are there	View History Page	We can see history	Working as expected	Pass	-	N	-	Abuthahir Nithish Kumar N
View History TC 02	Functional	View History Page	User is able to the nutritional breakdown of the previously uploaded images	Page is accessible	1. See if the nutritional breakdown is visible	View History Page	We can see the nutrients	Working as expected	Pass	-	N	-	Abuthahir Nithish Kumar N Nandita Sajeev

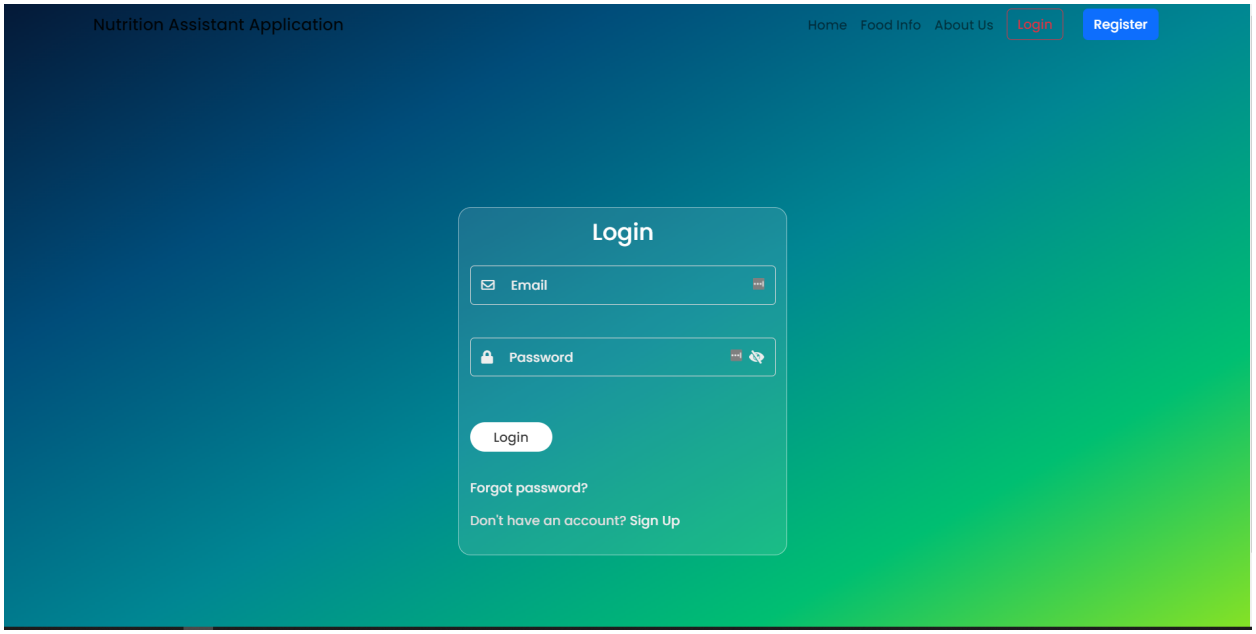
9. RESULTS

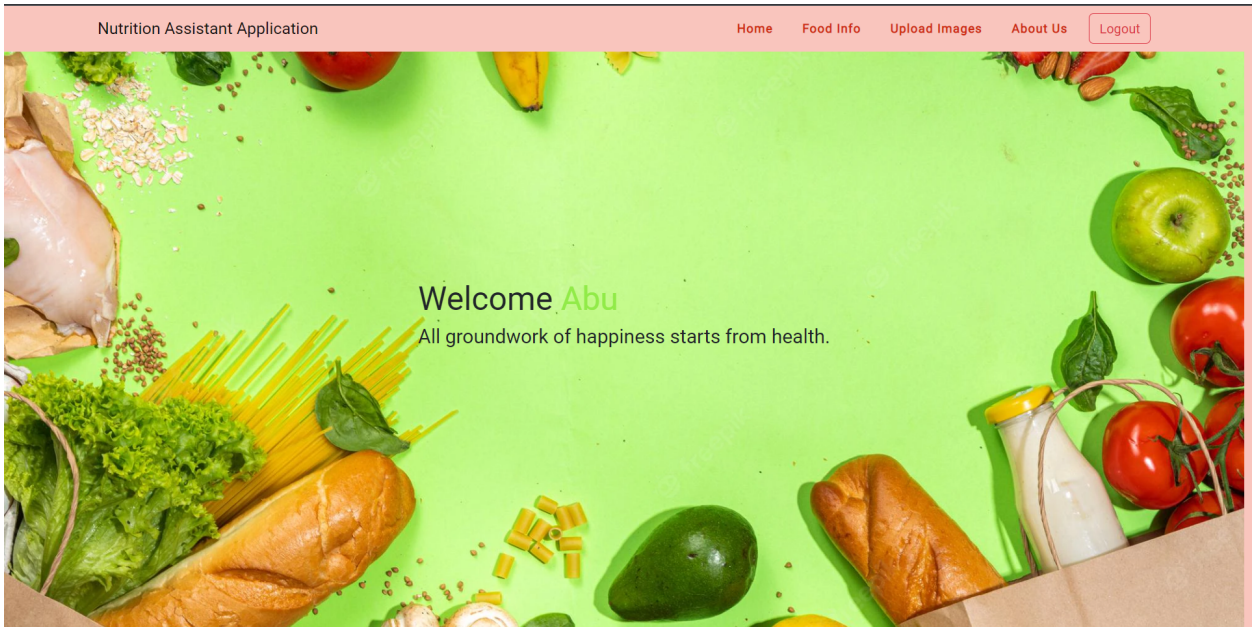
9.1 Performance Metrics

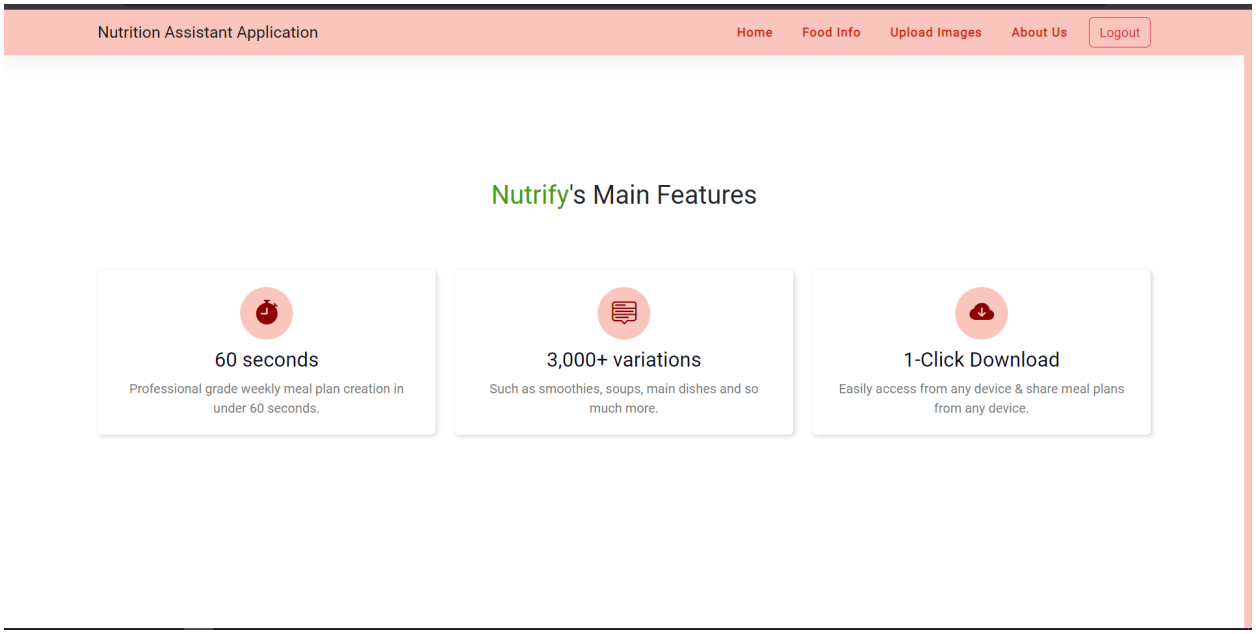
S.No	Parameters	Values	Screenshots
1	Homepage	This page allows the user to get a glimpse of the app and allows them to navigate the	
2	Upload Image Page	This page allows users to upload food images and get results	
3	View History Page	User is able to view the past uploaded items	
4	Login/SignUp	User can log in and sign up	

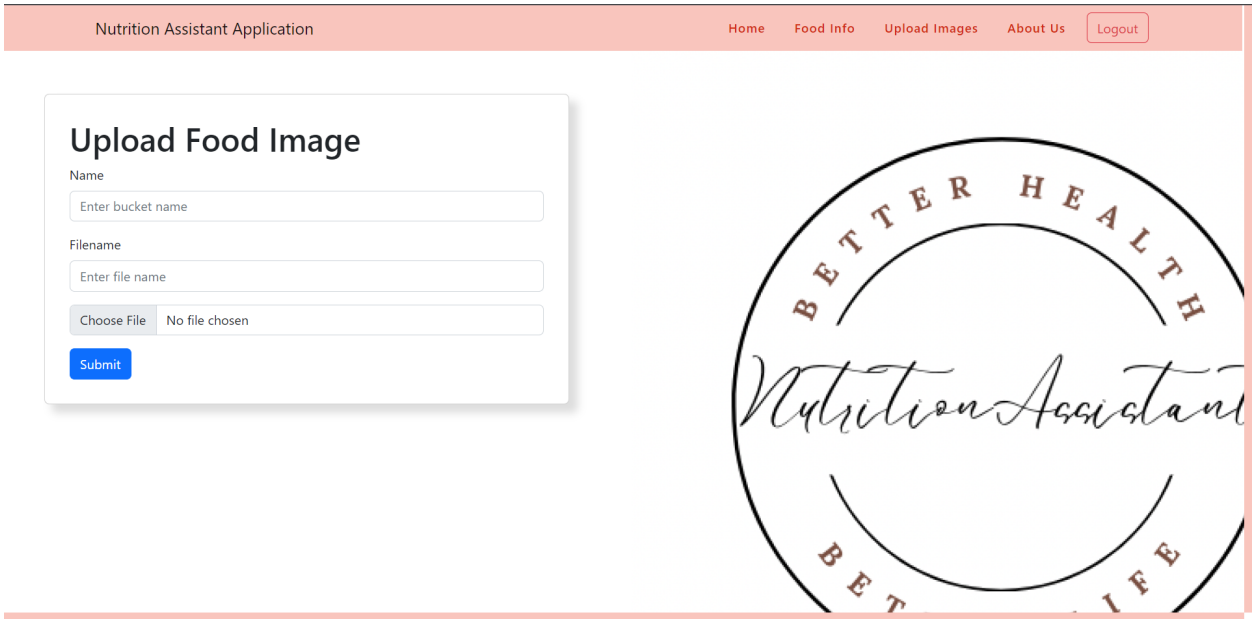
9.2 Screenshots of UI




Nutrition Assistant Application
Home
Food Info
About Us
Login
Register
Login
Email
Password
Login
Forgot password?
Don't have an account? Sign Up


Nutrition Assistant Application
Home
Food Info
Upload Images
About Us
Logout
Welcome Abu
All groundwork of happiness starts from health.


Nutrition Assistant Application
Home
Food Info
Upload Images
About Us
Logout
Nutrify's Main Features
60 seconds
Professional grade weekly meal plan creation in under 60 seconds.
3,000+ variations
Such as smoothies, soups, main dishes and so much more.
1-Click Download
Easily access from any device & share meal plans from any device.


Nutrition Assistant Application
Home
Food Info
Upload Images
About Us
Logout
Upload Food Image
Name
Enter bucket name
Filename
Enter file name
Choose File
No file chosen
Submit
BETTER HEALTH
Nutrition Assistant

Nutrition Assistant Application

Home Food Info Upload Images About Us Logout

Food item: agnolotti

Calories: 416.0 Kcal

Carbs: 18.0 Gm

Fat: 9.0 Gm

Protein: 6 Gm

Learn More..

10. ADVANTAGES AND DISADVANTAGES

Advantages:

1. The user is now able to track his daily calorie intake.
2. He/she can now take effective measures to maintain a healthy body weight.
3. It delivers information on the nutritional value of food and how it should be maintained on a daily basis.

Disadvantages:

1. It cannot be used without an Internet Connection.
2. Usage of 3rd party API may cause a time delay.

11. CONCLUSION

The nutrition assistant application using cloud computing is able to get images from the users and analyze them and show the nutritional breakdown of the food item. It is able to do this in an efficient and cost-effective way. This application allows people to get to know the nutrients of foods at any time which makes it more convenient for the users. This can be scaled to include APIs that have a larger variety of foods to have it cater to larger audiences of different backgrounds and ethnicities.

12. FUTURE SCOPE

The application can be improved to cater to more people

1. **ADDING GRAPHICAL DATA ON THE FOODS CONSUMES**
   Adding a pie chart or a breakdown of what nutritional components are being consumed can give more insight into the food habits of a user. This can help the user make changes and increase or decrease their consumption of a particular nutrient or food.

2. **CREATING A PERSONALISED FOOD RECOMMENDATION SYSTEM**
   Based on the previously uploaded images we can provide recommendations for the kinds of foods to eat to have a balanced diet.

13. APPENDIX

13.1 Source Code

home.html:

```
{% extends "base.html" %}

{% block head %}
        <title>Nutrify</title>
    <link rel="stylesheet" type="text/css" href="{{url_for('static',
filename='css/indexstyle.css')}}">
{% endblock %}

{% block body %}
<header>
    <nav class="navbar navbar-expand-lg navigation-wrap">
        <div class="container">
            <a class="navbar-brand" href="/">Nutrition Assistant Application</a>
            <button class="navbar-toggler" type="button" data-bs-toggle="collapse"
data-bs-target="#navbarNav"
                aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle
navigation">
                <span class="navbar-toggler-icon"></span>
            </button>
            <div class="collapse navbar-collapse" id="navbarText">

                <ul class="navbar-nav ms-auto mb-2 mb-lg-0">
                    <li class="nav-item">
                        <a class="nav-link" aria-current="page" href="/">Home</a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link" aria-current="page" href="/foodinfo">Food
Info</a>
                    </li>
                    {% if session['username'] %}
                    <li class="nav-item">
                        <a class="nav-link" aria-current="page" href="/uploader">Upload
Images</a>
                    </li>
                    {% endif %}
                    <li class="nav-item">
                        <a class="nav-link" aria-current="page" href="/#aboutus">About Us</a>
                    </li>
                    {% if session['username'] %}
                    <li>
                        <a class="btn btn-outline-danger ms-2" href="{{ url_for('logout')
}}">Logout</a>
                    </li>
                    {% else %}
                        <li>
```

```
                                    <a class="btn btn-outline-danger ms-2" href="/login"
role="button">Login</a>
                                </li>
                                <li>
                                    <a class="btn btn-primary ms-4" href="/signup"
role="button">Register</a>
                                </li>
                            {% endif %}
                        </ul>
                    </div>
                </div>
            </nav>
        </header>
        <section id="home">

          <div class="test">
            <div class="main">

            </div>
            <div class="textbox">
                {% if session['username'] %}
                <h1>Welcome <span style="color:#91eb48;">{{ session['username'] | title
}}</span></h1>
                <!-- <a href="{{ url_for('logout') }}">Logout</a> -->
            {% else %}
                <h1>Welcome to  <span style="color:#40a008;">Nutrify</span></h1>
            {% endif %}

            <h4>All groundwork of happiness starts from health.</h4>
            </div>
          </div>




        </section>

        <section class="feature section-padding" id="feature">
               <div class="container-fluid px-0 top-banner1">
               <div class="container">
                      <div class="feature-heading" style="visibility: visible; animation-name: zoom">
                             <h2><span style="color:#40a008;">Nutrify</span>'s Main Features</h2>
                      </div>
                      <div class="serv-field row mt-4">
                        <div class="col-12 col-md-6 col-lg-4 text-center">
                            <div class="serv-box">
                                <div class="icon">
                                    <i class="bi bi-stopwatch-fill"></i>
                                </div>
                                <h4>60 seconds</h4>
                                <p>Professional grade weekly meal plan creation in under 60
seconds.</p>
```

```
                        </div>
                    </div>
                    <div class="col-12 col-md-6 col-lg-4 text-center">
                        <div class="serv-box">
                            <div class="icon">
                                <i class="bi bi-menu-up"></i>
                            </div>
                            <h4>3,000+ variations</h4>
                            <p>Such as smoothies, soups, main dishes and so much more.</p>
                        </div>
                    </div>
                    <div class="col-12 col-md-6 col-lg-4 text-center">
                        <div class="serv-box">
                            <div class="icon">
                                <i class="bi bi-cloud-arrow-down-fill"></i>
                            </div>
                            <h4>1-Click Download</h4>
                            <p>Easily access from any device &amp;amp; share meal plans from any
device.</p>
                        </div>
                    </div>
                </div>
            </div>
        </div>
        </section>
        <script type="text/javascript" >
            let nav=document.querySelector(".navigation-wrap");
            window.onscroll = function(){
                if(document.documentElement.scrollTop > 20){
                    nav.classList.add("scroll-on");
                }else{
                    nav.classList.remove("scroll-on");
                }
            }

        </script>

        <section class="feature section-padding" id="trackfood">
            <div class="container-fluid px-0 top-banner1">
            <div class="container">
                <div class="feature-heading" style="visibility: visible; animation-name: zoom">
                    <h2>KNOW THE NUTRIENTS PRESENT IN YOUR FOOD</h2>
                </div>
                <div class="serv-field row mt-4">
                  <div class="col-12 col-md-6 col-lg-4 text-center">
                        <div class="serv-box">
                            <div class="icon">
                                <i class="bi bi-camera-fill"></i>
                            </div>
                           <h1 style="font-size: xxx-large;
             color: red;
             font-family: 'Gill Sans', 'Gill Sans MT', Calibri, 'Trebuchet MS',
sans-serif;">01</h1>
```

```
                            <p>Click a Picture of Your Food</p>
                        </div>
                    </div>
                    <div class="col-12 col-md-6 col-lg-4 text-center">
                        <div class="serv-box">
                            <div class="icon">
                                <i class="bi bi-cloud-arrow-up-fill"></i>
                            </div>
                           <h1 style="font-size: xxx-large;
                color: red;
                font-family: 'Gill Sans', 'Gill Sans MT', Calibri, 'Trebuchet MS',
sans-serif;">02</h1>
                            <p>Upload the Picture</p>
                        </div>
                    </div>
                    <div class="col-12 col-md-6 col-lg-4 text-center">
                        <div class="serv-box">
                            <div class="icon">
                                <i class="bi bi-card-list"></i>
                            </div>
                           <h1 style="font-size: xxx-large;
                color: red;
                font-family: 'Gill Sans', 'Gill Sans MT', Calibri, 'Trebuchet MS',
sans-serif;">03</h1>
                            <p>Know the Nutrients Present</p>
                        </div>
                    </div>
                </div>
                <br>
                <br>
                <center>
                    {% if session['username'] %}
            <a href="/uploader" class="btn btn-outline-primary" role="button" aria-pressed="true">
TRY NOW </a>
            {% endif %}
          </center>
              </div>
        </div>

        </section>


        <footer id="aboutus">
                <div class="container">
                    <div class="row pb-4">
                        <div class="foot-info col-12 col-md-6 col-lg-9">
                            <a class="foot-logo" href="#home">
                                Nutri<span style="color:#40a008;" >fy</span>
                            </a>
                            <div class="mail">
                                <i class="fas fa-envelope"></i>
```

```
                                <a
href="mailto:nandita.sajeev.2019.cse@rajalakshmi.edu.in">nandita.sajeev.2019.cse@rajalakshmi.edu.in</a
>
                            </div>
                            <div class="mail">
                               <p>Contact Us: +91 12345 67890</p>
                            </div>
                        </div>
                    </div>
            </footer>
        {% endblock %}
```

index.html:

```
        {% extends "base.html" %}
        {% block head %}
               <title>Nutrify</title>
            <link rel="stylesheet" type="text/css" href="{{url_for('static',
filename='css/indexstyle.css')}}">
                <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/css/bootstrap.min.css"
rel="stylesheet" integrity="sha384-Zenh87qX5JnK2Jl0vWa8Ck2rdkQ2Bzep5IDxbcnCeuOxjzrPF/et3URy9Bv1WTRi"
crossorigin="anonymous">
            <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/js/bootstrap.bundle.min.js"
integrity="sha384-OERcA2EqjJCMA+/3y+gxIOqMEjwtxJY7qPCqsdltbNJuaOe923+mo//f6V8Qbsw3"
crossorigin="anonymous"></script>
        {% endblock %}
        {% block body %}


              {% for row in files %}
                 <div class="piccard card">
                    <img class="pic" src="{{row[0]}}">
                    <a href="/nutritioninfo/{{row[1]}}"><button  style="width: 30%;" type="submit"
class="btn btn-outline-primary">Learn More..</button>
                    </a>
                 </div>

              {% endfor %}
              {%  endblock %}
```

upload.html:

```
        {% extends "base.html" %}

        {% block head %}
               <title>Nutrify</title>
            <link rel="stylesheet" type="text/css" href="{{url_for('static',
filename='css/indexstyle.css')}}">
                <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/css/bootstrap.min.css"
rel="stylesheet" integrity="sha384-Zenh87qX5JnK2Jl0vWa8Ck2rdkQ2Bzep5IDxbcnCeuOxjzrPF/et3URy9Bv1WTRi"
crossorigin="anonymous">
            <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/js/bootstrap.bundle.min.js"
```

```
integrity="sha384-OERcA2EqjJCMA+/3y+gxIOqMEjwtxJY7qPCqsdltbNJuaOe923+mo//f6V8Qbsw3"
crossorigin="anonymous"></script>
        {% endblock %}



        {% block body %}
        <div class="row">

               <div class="col-lg-6">
               <div class="upload card">
                      <h1>Upload Food Image</h1>
               <form class="uploaderform" action = "/uploader" method = "POST"
               enctype = "multipart/form-data">

                 <div class="mb-3">
                      <label for="exampleInputname" class="form-label">Name</label>
                      <input placeholder="Enter bucket name" type="text" name="bucket"
class="form-control" id="exampleInputname" aria-describedby="emailHelp">
                 </div>
                 <div class="mb-3">
                      <label for="exampleInputfilename" class="form-label">Filename</label>
                      <input placeholder="Enter file name" name="filename" type="text"
class="form-control" id="exampleInputfilename">
                 </div>
                 <div class="mb-3">
                        <input  class="form-control" type = "file" name = "file" />

                 </div>
                 <button type="submit" class="btn btn-primary">Submit</button>
               </form>
        </div>
        </div>

        <div class="col-lg-6">
        <img src="/static/images/logo.png" alt="">
        </div>

        </div>




        {%  endblock %}
```

foodinfo.html:

```
        {% extends "base.html" %}

        {% block head %}
               <title>Nutrify</title>
```

```
            <link rel="stylesheet" type="text/css" href="{{url_for('static',
filename='css/indexstyle.css')}}">
        {% endblock %}

        {% block body %}
        <br>
        <br>
        <br>

        <div class="foodinfomain">
        <div class="foodinfo card">
            <div class="row">
                <div class="col-lg-6">
                    <img class="fo" src="{{files[8]}}" alt="">

                </div>
        <div class="col-lg-6">

            <p>Food item: {{files[1]}}</p>
            <hr>
            <p>Calories: {{files[2]}} Kcal</p>
            <hr>
            <p>Carbs: {{files[3]}} Gm</p>
            <hr>
            <p>Fat: {{files[4]}} Gm</p>
            <hr>
            <p>Protein: {{files[5]}} Gm</p>

        </div>
            </div>

        </div>
        </div>



        {% endblock %}
```

register.html:

```
        {% extends "base.html" %}

        {% from "_render_field.html" import render_error_field %}

        {% block head %}
        <title>Registration</title>
        <link rel="stylesheet" href="{{url_for('static', filename='css/login_style.css')}}">
        {% endblock %}

        {% block body %}
           <div class="container h-100">
            <div class="wrapper">
                <div class="card">
```

```
                    <form class="d-flex flex-column" action="{{ url_for('signup') }}" method="POST"
novalidate>
                        {{ form.csrf_token }}
                        <div class="h3 text-center text-white">Sign Up</div>
                        <div class="d-flex align-items-center input-field my-3 mb-4">
                            <span class="far fa-user p-2"></span>
                            {{ form.username(class="form-control", placeholder="Username") }}
                        </div>
                        {{ render_error_field(form.username) }}

                        <div class="d-flex align-items-center input-field mb-4">
                            <span class="far fa-envelope p-2"></span>
                            {{ form.email(class="form-control", placeholder="Email") }}
                        </div>
                        {{ render_error_field(form.email) }}

                        <div class="d-flex align-items-center input-field mb-4">
                            <span class="fas fa-lock p-2"></span>
                            {{ form.pass1(class="form-control", placeholder="New Password", id='pwd')
}}
                            <button type="button" class="btn" onclick="showPassword()">
                                <span class="fas fa-eye-slash"></span>
                            </button>
                        </div>
                        {{ render_error_field(form.pass1) }}

                        <div class="d-flex align-items-center input-field mb-4">
                            <span class="fas fa-lock p-2"></span>
                            {{ form.pass2(class="form-control", placeholder="New Password",
id='pwd1') }}
                            <button type="button" class="btn" onclick="Password()">
                                <span class="fas fa-eye-slash"></span>
                            </button>
                        </div>
                        {{ render_error_field(form.pass2) }}

                        <div class="my-3">
                            <input type="submit" value="Register" class="btn btn-primary">
                        </div>

                        <div class="mb-3">
                            <span class="text-light-white">Already have an account?</span>
                            <a href="/login">Login</a>
                        </div>

                    </form>

                </div>
            </div>
        </div>
        {% endblock %}
```

login.html:

```
{% extends "base.html" %}

{% from "_render_field.html" import render_error_field %}

{% block head %}
<title>Login</title>
<link rel="stylesheet" href="{{url_for('static', filename='css/login_style.css')}}">
{% endblock %}


{% block body %}
<div class="container h-100">
    <div class="wrapper">
        <div class="card">
            <form class="d-flex flex-column" method="POST" action="{{ url_for('login') }}"
novalidate >
                {{ form.csrf_token }}
                <div class="h3 text-center text-white">Login</div>
                <div class="d-flex align-items-center input-field my-3 mb-4">
                    <span class="far fa-envelope p-2"></span>
                    {{ form.email(class="form-control", placeholder="Email") }}
                    <!-- <input type="text" placeholder="Email" required
class="form-control"> -->
                </div>
                {{ render_error_field(form.email) }}
                <div class="d-flex align-items-center input-field mb-4">
                    <span class="fas fa-lock p-2"></span>
                    {{ form.password(class="form-control", id="pwd", placeholder="Password")
}}
                    <!-- <input type="password" placeholder="Password" required
class="form-control" id="pwd"> -->
                    <button type="button" class="btn" onclick="showPassword()">
                        <span class="fas fa-eye-slash"></span>
                    </button>
                </div>
                {{ render_error_field(form.password) }}
                <div class="my-3">
                    <input type="submit" value="Login" class="btn btn-primary">
                </div>
                <div class="d-sm-flex align-items-sm-center justify-content-sm-between my-3">
                    <div class="mt-sm-0 mt-3"><a href="/forgot">Forgot password?</a></div>
                </div>

                <div class="mb-3">
                    <span class="text-light-white">Don't have an account?</span>
                    <a href="/signup">Sign Up</a>
                </div>
            </form>

        </div>
    </div>
</div>
```

```
{% endblock %}
```

forgot_password.html:

```
{% extends "base.html" %}

{% from "_render_field.html" import render_error_field %}

{% block head %}
<title>Reset Password</title>
<link rel="stylesheet" href="{{url_for('static', filename='css/reset_style.css')}}">
{% endblock %}

{% block body %}
<div class="container h-100">
    <div class="wrapper">
        <div class="card">
            <form action="{{ url_for('forgot_password') }}" method="POST" class="d-flex
flex-column" novalidate>
                {{ form.csrf_token }}
                <div class="h3 text-center text-white">Reset Password</div>
                <div class="d-flex align-items-center input-field my-4 mb-4">
                    <span class="far fa-envelope p-2"></span>
                    {{ form.email(class="form-control", placeholder="Registered Email") }}
                </div>
                {{ render_error_field(form.email) }}
                <div class="my-3">
                    <input type="submit" value="Reset" class="btn btn-primary">
                </div>

            </form>

        </div>
    </div>
</div>
{% endblock %}
```

base.html:

```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <!-- Bootstrap -->
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css"
rel="stylesheet"
        integrity="sha384-EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTwFspd3yD65VohhpuuCOmLASjC"
crossorigin="anonymous">
    <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js"
```

```
                integrity="sha384-MrcW6ZMFYlzcLA8Nl+NtUVF0sA7MsXsP1UyJoMp4YLEuNSfAP+JcXn/tWtIaxVXM"
                crossorigin="anonymous"></script>
            <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/css/bootstrap.min.css"
rel="stylesheet">
            <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/js/bootstrap.bundle.min.js"></script>

            <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.5.0/font/bootstrap-icons.css" />
            <link rel="stylesheet" href="https://use.fontawesome.com/releases/v5.3.1/css/all.css"
                integrity="sha384-mzrmE5qonljUremFsqc01SB46JvROS7bZs3IO2EmfFsd15uHvIt+Y8vEf7N7fWAU"
crossorigin="anonymous">

            <!-- Fonts -->
            <link href="https://fonts.googleapis.com/css2?family=Poppins:wght@300&display=swap"
rel="stylesheet">
            <link href="https://fonts.googleapis.com/css2?family=Poppins:wght@500&display=swap"
rel="stylesheet">
            <link rel="stylesheet" href="https://use.fontawesome.com/releases/v5.3.1/css/all.css"
                integrity="sha384-mzrmE5qonljUremFsqc01SB46JvROS7bZs3IO2EmfFsd15uHvIt+Y8vEf7N7fWAU"
crossorigin="anonymous">
            <!-- CSS -->
            <link rel="stylesheet" href="{{url_for('static', filename='css/navbar.css')}}">
            <!-- <link rel="stylesheet" href="{{url_for('static', filename='css/style.css')}}"> -->

            <!-- JS Script for-->
            <script type="text/javascript">

                function showPassword() {
                    var password = document.getElementById('pwd');
                    if (password.type === 'password') {
                        password.type = "text";
                    }
                    else {
                        password.type = "password";
                    }
                }

                function Password() {
                    var password = document.getElementById('pwd1');
                    if (password.type === 'password') {
                        password.type = "text";
                    }
                    else {
                        password.type = "password";
                    }
                }

            </script>
                {% block head %} {% endblock %}

        </head>
```

```
        <body>
            {% with messages = get_flashed_messages(category_filter=["error"]) %}
            {% if messages %}
            <div class="alert alert-danger" role="alert">
                <ul class=flashes>
                    {% for message in messages %}
                    <li>{{ message }}</li>
                    {% endfor %}
                </ul>
            </div>
            {% endif %}
            {% endwith %}

            {% with messages = get_flashed_messages(category_filter=["success"]) %}
            {% if messages %}
            <div class="alert alert-success" role="alert">
                <ul class=flashes>
                    {% for message in messages %}
                    <li>{{ message }}</li>
                    {% endfor %}
                </ul>
            </div>
            {% endif %}
            {% endwith %}

        <header>
            <nav class="navbar navbar-expand-lg navigation-wrap">
                <div class="container">
                    <a class="navbar-brand" href="/">Nutrition Assistant Application</a>
                    <button class="navbar-toggler" type="button" data-bs-toggle="collapse"
data-bs-target="#navbarNav"
                        aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle
navigation">
                        <span class="navbar-toggler-icon"></span>
                    </button>
                    <div class="collapse navbar-collapse" id="navbarText">

                        <ul class="navbar-nav ms-auto mb-2 mb-lg-0">
                            <li class="nav-item">
                                <a class="nav-link" aria-current="page" href="/">Home</a>
                            </li>

                            <li class="nav-item">
                                <a class="nav-link" aria-current="page" href="/foodinfo">Food
Info</a>
                            </li>
                            {% if session['username'] %}
                            <li class="nav-item">
                                <a class="nav-link" aria-current="page" href="/uploader">Upload
Images</a>
                            </li>
                            {% endif %}
```

```
                            <li class="nav-item">
                                <a class="nav-link" aria-current="page" href="/#aboutus">About Us</a>
                            </li>
                            {% if session['username'] %}
                            <li>

                                <a class="btn btn-outline-danger ms-2" href="{{ url_for('logout')
}}">Logout</a>
                            </li>
                            {% else %}
                                <li>
                                    <a class="btn btn-outline-danger ms-2" href="/login"
role="button">Login</a>
                                </li>
                                <li>
                                    <a class="btn btn-primary ms-4" href="/signup"
role="button">Register</a>
                                </li>
                            {% endif %}
                        </ul>
                    </div>
                </div>
            </nav>
        </header>


            {% block body %} {% endblock %}

        </body>

        </html>
```

delete.html:

```
        <html>

           <body>

                <a href="/">HOME</a>
                <a href="/uploader">Upload </a>
                <a href="/deletefile">Delete </a>
                <br><hr>

        <h1>IBM Object Storage</h1>
        <div>
              <form action = "/deletefile" method = "POST" >

                        <input type = "text" placeholder="Enter bucket name" name = "bucket" />
                        <br>
                        <br>
                        <input type = "text" placeholder="Enter file name" name = "filename" />
                        <br>
                        <br>
                  <input type = "submit"/>
```

```
      </form>
       </div>
   </body>
</html>
```

_render_field:

```
{% macro render_error_field(field) %}
<ul>
    {% for error in field.errors %}
        <li class="field_error">{{ error }}</li>
    {% endfor %}
</ul>
{% endmacro %}
```

indexstyle.css:

```
body, html {
    height: 100%;
  }
a,
a:hover{
       text-decoration: none;
}

a:hover{
       color: #FF0000;
}

html{
       scroll-behavior: smooth;
}

body{
       font-family: 'Roboto', sans-serif;
       font-size: 100%;
       font-weight: 400;
}

::-webkit-scrollbar{
       width: 0.625rem;
}

::-webkit-scrollbar-track{
       background: #f9c5bd;
}

::-webkit-scrollbar-thumb{
       background: #f9c5bd;
}

.navigation-wrap{
```

```
    background-color:#f9c5bd;
       position: fixed;
       width: 100%;
       left:0;
       z-index: 1000;
       -webkit-transition:all 0.3s ease-out;
       transition: all 0.3s ease-out;
}

.navigation-wrap .nav-item{
       padding: 0 0.625rem;
       transition: all 200ms linear;
}

.navbar-toggler:focus{
       outline: unset;
       border: unset;
       box-shadow: none;
}

.nav-item .nav-link{
       font-size: 0.9375rem;
       font-weight: 600;
       text-transform: capitalize;
       color: #D12B10;
       letter-spacing: 1px;
}
.nav-item .nav-link a:hover{
       color: #FF0000;
}

.navigation-wrap.scroll-on{
       position: fixed;
       top:0;
       left: 0;
       width: 100%;
       background: #f9c5bd;
       box-shadow: 0 0.125rem 1.75rem 0 rgb(0,0,0,0.09);
       transition: all .15s ease-in-0ut 0s;
}

#home{
    height: 100%;
}

.test{
    height: 100%;
    position: relative;
}

.top-banner{
       width: 80%;
```

```
        padding: 10rem 0 7rem;

}

.main{
  background-image: url('/static/images/home.webp');
  height: 100%;
  width: 100%;
  font-size: 70px;
  background-position: center;
  background-repeat: no-repeat;
  background-size: cover;
}

.textbox{
    position: absolute;
    top: 50%;
    left: 50%;
    transform: translate(-50%, -50%);

}

.upload{
    padding: 4%;
    margin:7%;

    box-shadow: 10px 10px 10px #ddd;
}

.piccard{
width: 30%;
padding: 2%;
border-radius: 7rem;
}

.pic{
    margin-bottom: 7%;
}

.foodinfomain{

padding: 3%;
}


.foodinfo{
    text-align: right;
    position: absolute;
  top: 20%;
  left: 25%;
    padding: 2%;
    width:50%;
   border-radius: 2rem;
```

```
   box-shadow: 10px 10px 10px #ddd;
   font-size: 20px;


}
.fo{
    width: 130%;
}

.top-banner h1{
       font-size: 48px;
}

.top-banner1{
       width: 100%;
       padding: 9.875rem 0 7.375rem;
}

.feature-heading{
       text-align: center;
}

.serv-field {
    padding-top: 50px;
}
.serv-field .serv-box {
    padding: 20px;
    margin-bottom: 20px;
    border-radius: 5px;
    border: 1px solid transparent;
    box-shadow: 2px 2px 5px rgba(136, 136, 136, 0.3);
    transition:  0.4s ease
}
.serv-field .serv-box:hover {
    box-shadow: rgba(0, 0, 0, 0.4) 0px 30px 90px;
    border-color:darkred;
}
.serv-field .serv-box .icon {
    position: relative;
    width: 65px;
    height: 65px;
    display: flex;
    align-items: center;
    justify-content: center;
    margin: 0 auto 10px;
    color:darkred;
    border-radius: 50%;
}
.serv-field .serv-box .icon:before {
    content: '';
    position: absolute;
    top: 50%;
    left: 50%;
```

```
    transform: translate(-50%,-50%) scale(1);
    width: 100%;
    height: 100%;
    border-radius: 50%;
    background-color: #f9c5bd;
    z-index: -1;
}
.serv-field .serv-box .icon i {
    font-size: 30px;
}
.serv-field .serv-box h4 {
    font-size: 25px;
    color:  #090719;
}
.serv-field .serv-box p {
    color:  #787878;
    margin-bottom: 0px;
}

footer {
    background-color: #222;
    color: #fff;
    padding-top: 50px;
}
footer a {
   color: #fff;
    transition: color 0.4s ease;
    text-decoration: none;
}
footer a:hover{
 color: #D12B10;
}
footer ul {
    padding-left: 0;
}
footer .foot-logo {
    display: inline-block;
    font-size: 35px;
    font-weight: 700;

    margin-bottom: 15px;
}
footer .foot-info > div {
    display: flex;
    margin-bottom: 15px;
}
footer .foot-info > div i {
    width: 35px;
    font-size: 18px;
}
```

login_style.css:

```
* {
    margin: 0;
    padding: 0;
    box-sizing: border-box;
    font-family: 'Poppins', sans-serif;
}

body {
   background-image: linear-gradient(to right bottom, #051937, #004d7a, #008793, #00bf72,
#a8eb12);
    background-repeat: no-repeat;
    height: auto;
}

.wrapper {
    max-width: 500px;
    margin-top: 15%;
    margin-left: 32%;
    margin-bottom: 13.5%;
}

.wrapper .card {
    max-width: 400px;
    min-height: 380px;
    margin: 30px;
    background: rgba(255, 255, 255, 0.1);
    overflow: hidden;
    backdrop-filter: blur(10px);
    border: 1px solid rgba(255, 255, 255, 0.5);
    border-radius: 15px;
    cursor: pointer;
    padding: 0.8rem;
}

.wrapper .card a {
    text-decoration: none;
    color: #eee;
}

.wrapper .card a:hover {
    color: #fff;
}

.wrapper .card .input-field {
    border: 1px solid #ddd;
    border-radius: 5px;
    color: #eee;
    padding: 0.3rem;
}

.wrapper .card .input-field input {
    background-color: inherit;
```

```
}

.wrapper .card .input-field input.form-control,
.wrapper .card .input-field input.form-control:focus {
    border: none;
    outline: none;
    box-shadow: none;
    color: #eee;
}

.wrapper .card .input-field button.btn {
    color: #eee;
    padding: 0rem;
    padding-right: 0.5rem;
}

.wrapper .card .input-field button.btn:hover {
    color: #fff;
}

.wrapper .card .input-field button.btn:focus {
    border: none;
    outline: none;
    box-shadow: none;
}

.wrapper .card .input-field input::placeholder {
    color: #eee;
}

.wrapper .card .option {
    display: block;
    position: relative;
    padding-left: 25px;
    cursor: pointer;
    user-select: none
}

.wrapper .card .option span.text-light-white:hover {
    color: #fff;

}

.wrapper .card .option input {
    position: absolute;
    opacity: 0;
    cursor: pointer;
    height: 0;
    width: 0
}

.checkmark {
    position: absolute;
```

```
    top: 3px;
    left: 0;
    height: 18px;
    width: 18px;
    background-color: #fff;
    border-radius: 2px
}



.wrapper .card .btn.btn-primary {
    border-radius: 20px;
    width: 100px;
    background-color: #fff;
    color: #333;
    border: none;
}

.wrapper .card .btn.btn-primary:hover {
    color: #fff;
    background: #333;
}

.wrapper .card .btn.btn-primary:focus {
    border: none;
    box-shadow: none;
}

.wrapper .card .text-light-white {
    color: #ddd;
}

.wrapper .card .line span.connect {
    position: absolute;
    top: -12px;
    left: 33%;
    color: #000;
    padding: 0 0.3rem;
    z-index: 100;
    border-radius: 2px;
    background-color: #fff;
}

.wrapper .card .connections a img {
    width: 40px;
    height: 40px;
    border-radius: 50%;
    object-fit: cover;

}
```

navbar.css:

```
a,
a:hover{
	text-decoration: none;
}

a:hover{
	color: #FF0000;
}

html{
	scroll-behavior: smooth;
}

body{
	font-family: 'Roboto', sans-serif;
	font-size: 100%;
	font-weight: 400;
}
```

reset_style.css:

```
* {
    margin: 0;
    padding: 0;
    box-sizing: border-box;
    font-family: 'Poppins', sans-serif;
}

body {
   background-image: linear-gradient(to right bottom, #051937, #004d7a, #008793, #00bf72,
#a8eb12);
    background-repeat: no-repeat;
    height: auto;
}

.wrapper {
    max-width: 500px;
    margin-top: 17%;
    margin-left: 32%;
    margin-bottom: 24%;
}

.wrapper .card {
    max-width: 400px;
    min-height: 200px;
    margin: 30px;
    background: rgba(255, 255, 255, 0.1);
    overflow: hidden;
    backdrop-filter: blur(10px);
    border: 1px solid rgba(255, 255, 255, 0.5);
    border-radius: 15px;
```

```
    cursor: pointer;
    padding: 0.8rem;
}

.wrapper .card a {
    text-decoration: none;
    color: #eee;
}

.wrapper .card a:hover {
    color: #fff;
}

.wrapper .card .input-field {
    border: 1px solid #ddd;
    border-radius: 5px;
    color: #eee;
    padding: 0.3rem;
}

.wrapper .card .input-field input {
    background-color: inherit;
}

.wrapper .card .input-field input.form-control,
.wrapper .card .input-field input.form-control:focus {
    border: none;
    outline: none;
    box-shadow: none;
    color: #eee;
}

.wrapper .card .input-field button.btn {
    color: #eee;
    padding: 0rem;
    padding-right: 0.5rem;
}

.wrapper .card .input-field button.btn:hover {
    color: #fff;
}

.wrapper .card .input-field button.btn:focus {
    border: none;
    outline: none;
    box-shadow: none;
}

.wrapper .card .input-field input::placeholder {
    color: #eee;
}

.wrapper .card .option {
```

```
    display: block;
    position: relative;
    padding-left: 25px;
    cursor: pointer;
    user-select: none
}

.wrapper .card .option span.text-light-white:hover {
    color: #fff;

}

.wrapper .card .option input {
    position: absolute;
    opacity: 0;
    cursor: pointer;
    height: 0;
    width: 0
}

.checkmark {
    position: absolute;
    top: 3px;
    left: 0;
    height: 18px;
    width: 18px;
    background-color: #fff;
    border-radius: 2px
}



.wrapper .card .btn.btn-primary {
    border-radius: 20px;
    width: 100px;
    background-color: #fff;
    color: #333;
    border: none;
}

.wrapper .card .btn.btn-primary:hover {
    color: #fff;
    background: #333;
}

.wrapper .card .btn.btn-primary:focus {
    border: none;
    box-shadow: none;
}

.wrapper .card .text-light-white {
    color: #ddd;
}
```

```
.wrapper .card .line span.connect {
    position: absolute;
    top: -12px;
    left: 33%;
    color: #000;
    padding: 0 0.3rem;
    z-index: 100;
    border-radius: 2px;
    background-color: #fff;
}

.wrapper .card .connections a img {
    width: 40px;
    height: 40px;
    border-radius: 50%;
    object-fit: cover;

}



            .nav-link{
     font-family: 'Bungee Spice', Georgia;
    font-size: 20px;
    padding-top: 29px;
    color: rgb(127,255,0);
}

.nav-link:hover{
    color: #00FF00!important;
}
```

13.2 GitHub and Project Demo Link

Github : https://github.com/IBM-EPBL/IBM-Project-18366-1659684009
Demo Video :
https://drive.google.com/file/d/1GAcdECwF7gC6cO8u5s4alS5hSs7zGkiB/view?usp=share_link